Ladner R., Warner, E., Petry, P., Gupta, K.M., Moore, P.G., Aha, D.W., & Shaw, K., (2006). Case-based classification alternatives to ontologies for automated web service discovery and integration, Proceedings of SPIE Defense & Security Symposium, (pp. 620117-1 – 620117-8), Vol. 6201

Case-Based Classification Alternatives to Ontologies for Automated Web Service Discovery and Integration

Roy Ladner[a], Elizabeth Warner[a], Fred Petry[a],
Kalyan Moy Gupta[b], Philip Moore[c], David W. Aha,[d] Kevin Shaw[a]
[a]Naval Research Laboratory, Stennis Space Center, MS
[b]Knexus Research, Springfield, VA, [c]ITT Industries, Alexandria, VA
[d]Naval Research Laboratory, Washington, DC

ABSTRACT

Web Services are becoming the standard technology used to share data for many Navy and other DoD operations. Since Web Services technologies provide for discoverable, self-describing services that conform to common standards, this paradigm holds the promise of an automated capability to obtain and integrate data. However, automated integration of applications to access and retrieve data from heterogeneous sources in a distributed system such as the Internet poses many difficulties. Assimilation of data from Web-based sources means that differences in schema and terminology prevent simple querying and retrieval of data. Thus, machine understanding of the Web Services interface is necessary for automated selection and invocation of the correct service. Service availability is also an issue that needs to be resolved. There have been many advances on ontologies to help resolve these difficulties to support the goal of sharing knowledge for various domains of interest.

In this paper we examine the use of case-based classification as an alternative/supplement to using ontologies for resolving several questions related to knowledge sharing. While ontologies encompass a formal definition of a domain of interest, case-based reasoning is a problem solving methodology that retrieves and reuses decisions from stored cases to solve new problems, and case-based classification involves applying this methodology to classification tasks. Our approach generalizes well in sparse data, which characterizes our Web Services application. We present our study as it relates to our work on development of the Advanced MetOc Broker, whose objective is the automated application integration of meteorological and oceanographic (MetOc) Web Services.

Keywords: Web Services, Ontology, Case-based Classification, MetOc, Service Oriented Architecture, Automated Web, Semantic Web

INTRODUCTION

Information is now as important as tanks, ships and aircraft in today's military. Rapid access to data and the ability to share data are seen as significant to gaining superiority over opposing forces [1]. Web Services are becoming the technology used to share data for many Navy and other DoD operations. Web Services technologies provide access to discoverable, self-describing services that conform to common standards. Thus, this paradigm holds the promise of an automated capability to obtain and integrate data. However, the automated integration of applications to access and retrieve data from heterogeneous sources in a distributed system such as the Internet poses many difficulties. Assimilation of data from Web-based sources means that differences in schema and terminology prevent simple querying and retrieval of data. Machine understanding of the Web Services interface is necessary for automated identification, selection and invocation of the correct service. Service availability must also be resolved.

There has been considerable work on ontologies to help resolve these difficulties so as to share knowledge among various domains of interest. Ontologies describe a formal definition for a domain of interest through the terms and concepts of the domain and their interrelationships, and support automated computer reasoning on a domain through

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Case-Based Classification Alternatives to Ontologies for Automated Web Service Discovery and Integration	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Knexus Research Corp,9120 Beachway Lane,Springfield,VA,22153**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Web Services are becoming the standard technology used to share data for many Navy and other DoD operations. Since Web Services technologies provide for discoverable, self-describing services that conform to common standards, this paradigm holds the promise of an automated capability to obtain and integrate data. However, automated integration of applications to access and retrieve data from heterogeneous sources in a distributed system such as the Internet poses many difficulties. Assimilation of data from Web-based sources means that differences in schema and terminology prevent simple querying and retrieval of data. Thus, machine understanding of the Web Services interface is necessary for automated selection and invocation of the correct service. Service availability is also an issue that needs to be resolved. There have been many advances on ontologies to help resolve these difficulties to support the goal of sharing knowledge for various domains of interest. In this paper we examine the use of case-based classification as an alternative/supplement to using ontologies for resolving several questions related to knowledge sharing. While ontologies encompass a formal definition of a domain of interest, case-based reasoning is a problem solving methodology that retrieves and reuses decisions from stored cases to solve new problems, and case-based classification involves applying this methodology to classification tasks. Our approach generalizes well in sparse data, which characterizes our Web Services application. We present our study as it relates to our work on development of the Advanced MetOc Broker, whose objective is the automated application integration of meteorological and oceanographic (MetOc) Web Services.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**8**	

specification of its content. In some uses of ontologies, Web Services data providers are presupposed to deploy an ontological description of their Web Service to support automated discovery and integration by interested client applications [2]. Our approach does not require such descriptions.

There has been some research on Web Service classification as a means of automating or semi-automating the annotation of Web Services with semantic meaning. That work has had as its focus the automatic generation of Web Services ontologies such as OWL-S [3, 4].

In this paper we depart from the use of ontologies and examine the direct use of case-based classification as an alternate approach to support automated discovery of meteorological and oceanographic Web Services. Case-based reasoning (CBR) is a problem solving methodology that retrieves and reuses decisions from stored cases to solve new problems, and case-based classification focuses on applying CBR to supervised classification tasks. This approach generalizes well in sparse data, which characterizes our Web Services application. Unlike ontologies, case-based classification does not require formal domain definition and its use does not require data providers to deploy any additional specialized descriptions of their Web Service.

We are currently developing an Advanced MetOc Broker (AMB) for the US Navy. Its objective is the automated discovery and application integration of meteorological and oceanographic (MetOc) Web Services. We are examining the use of case-based classification in the AMB to support automated Web Services discovery.

The remainder of this paper is organized as follows. First, we briefly overview Web Services and previous work on ontologies in support of automated data exchange. Following this, we describe our work on the AMB. We then explain our approach for classifying MetOc Web Services using a case-based classifier. We close with a discussion of future research goals.

WEB SERVICES AND ONTOLOGIES

Web Services provide data and services to users and applications over the Internet through a consistent set of standards and protocols such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), the Web Services Definition Language (WSDL), and Universal Discovery Description and Integration (UDDI). XML has become one of the widely used standards in interoperable exchange of data on the Internet but does not define the semantics of the data it describes. XML Schemas define XML documents through structures that describe elements, attributes and data types, among others [5]. WSDL describes the acceptable requests that will be honored by a Web Service, the types of responses that will be generated [6], and the XML messaging mechanism of the service. For example, the messaging mechanism may be specified as SOAP. A UDDI registry provides a way for data providers to advertise their Web Services and for consumers to find data providers and desired services. An interface to a UDDI registry may allow users to search for Web Services by business category, business name, or service [7]. This advertisement of Web Services may not be desirable for net-centric operations in the DoD community.

Interacting with multiple Web Service interfaces poses issues for client application integration and maintenance. Addressing these issues may involve adoption of a single, uniform Web Service interface that may be implemented by multiple diverse data providers within a community. Alternatively, when the data providers' interfaces are not uniformly defined, approaches such as that used by the Geospatial Information DataBase (GIDB®) Portal System may be advantageous. The GIDB is a Web-based portal service that allows users to access over 1500 data servers through a single graphical user interface. The GIDB accomplishes this using the data provider's interface of choice, which may be a Web Services interface, a native API, an Open Geospatial Consortium (OGC) Catalog, or another mechanism. Where the data provider's interface does not conform to a recognized standard, custom coding of the interface is necessary.

Recent efforts to improve interoperability include Web Services technologies such as WSDL and XML Schemas. While these provide structured content, their semantics are limited and not designed for interoperability (i.e., they may employ different meanings for the same terms or the same meanings using different terms, each of which limits their interoperability). Ontologies are often considered to be the basis of semantic meaning for these sorts of documents. Ontologies define the terms and concepts used to represent knowledge in a given domain of interest. They provide the structures that capture the relationships among concepts and enable applications to reason over them. Ontological frameworks for describing the semantics of data include such developments as the Resource Description Framework (RDF) and Web Ontology Language (OWL). RDF provides a flexible representation of information and a reliable means of supporting machine reasoning [8]. OWL permits users to more fully describe the meanings of terms found in Web documents and to represent the relationships among these terms [9].

Numerous methodologies for engineering and maintaining domain ontologies have been reported [10]. In some approaches, the starting point for ontology development is the specification of the questions the ontology should answer and/or problems it should solve. Generally, strategies for domain knowledge acquisition may vary from bottom-up to top-down. There are also editors that assist with ontology development, such as the open source editor Protégé. A Protégé extension supports OWL ontologies [11]. Even with these tools, ontology development remains a time- and skill-intensive activity.

OWL-S extends OWL to supply the constructs for defining an ontology of services that is intended to support automated Web Services discovery, invocation, and composition. For example, a Web Services provider could advertise its services in OWL-S in a service registry, where software agents or brokers could discover it through querying. The software agent or broker would then be able to interpret the OWL-S markup to determine whether the service provides the capability it needs, to understand the input required to invoke the service, and to determine what information will be returned. This is accomplished in the OWL-S ontology through classes that describe what the service does (service profile), how to ask for the service, what happens when the service is carried out (service grounding), and how the service can be accessed (service model) [12].

ADVANCED METOC BROKER

Our work on the AMB is focused on automated discovery and application integration of MetOc Web Services. We are engineering the AMB to automatically discover MetOc Web Services and dynamically translate data and methods across them. The AMB's Web Service search and discovery function is illustrated in Figure 1. We are developing the AMB to search identified registries for MetOc Web Services using the search feature supplied by that registry.

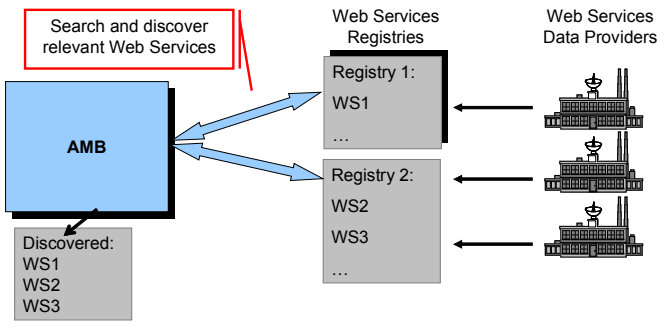

Fig. 1. The AMB search and discovery function.

The AMB's mediation function is depicted in Figure 2. We are developoing it to dynamically translate user requests to differing Web Service interface specifications. For example, this shall assist with brokering requests to multiple MetOc data providers whose services may have implemented a) a community standard interface, b) an interface that is not a community standard, or c) an evolving version of a community standard interface.

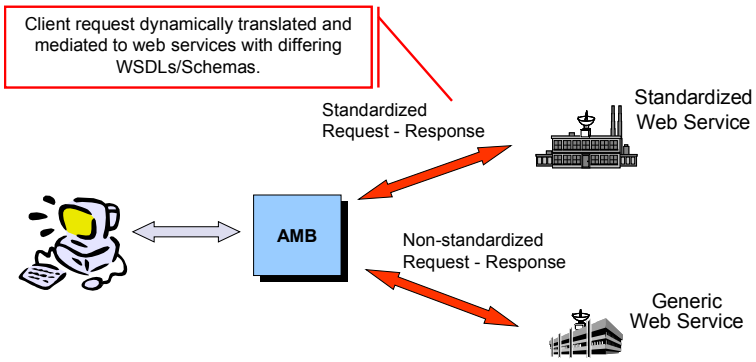

Fig. 2. The AMB mediation function.

While we are investigating the use of domain ontologies to automate the AMB, some of the AMB tasks seem suitable for resolution by automated classification techniques. One benefit of these techniques is that they do not require formal domain definition. More importantly, an automated classification approach does not rely on a data provider's deployment of additional specialized ontological descriptions of their Web Service, which is often lacking.

Identifying whether a particular Web Service supplies MetOc data can be framed as a classification task, which involves assigning one or more predefined labels to an unlabelled object. Thus, the Web Service identification task involves assigning the label "MetOc" or "Non-MetOc" to a given Web Service.

CASE-BASED CLASSIFICATION

Overview

Table 1: Example Web Services data for classifier learning

		Attributes -***A***					
		Conditional Attributes -***C***					Decision Attribute ***D***
		c_1 (zipcode)	c_2 (temperature)	c_3 (water)	c_4 (price)	c_5 (get)	d
O	o_1	3	2	1	0	1	Metoc
	o_2	1	0	0	2	3	Non-Metoc
	o_3	1	1	0	2	3	Non-Metoc
	o_4	2	1	4	1	4	Metoc

Our goal is to automatically build classifiers from example data, which is a *supervised learning* task. To formally describe a supervised classifier learning approach, we first present relevant notation. The example data required for our classifier learning algorithm must be in tabular form, where each row is an object o and each column is an attribute a (see Table 1). Let ***O*** represent the table's objects and ***A*** represent its attributes (i.e., columns). Each cell in the table is the value $v_{i,j}$ of the attribute for $\boldsymbol{a}_j$ for a particular object o_i. We partition the attributes into two types: (1) *Descriptor attributes* denoted by ***D,*** which are the object characteristics that provide information for classification, and (2) *Class attribute(s)* ***C,*** whose values indicate the category/class that applies to an object.

Learning a classifier implies inducing a function h that maps objects in $\boldsymbol{O}$ to classes in $\boldsymbol{C}$, that is, h: $\boldsymbol{O} \rightarrow \boldsymbol{C}$. The methods for estimating or learning h depend on the family of functions under consideration. For example, linear and non-linear regression techniques, back propagation with multilayer neural networks, top-down methods for inducing decision trees, support vector machines, and nearest neighbor rules are some of the methods used for inducing classifiers [13]. Different classifiers have different strengths and weaknesses depending on the characteristics of the example data and the target concept description. For example, statistical linear and non-linear regression techniques for classifier parameter estimation require a relatively large number of example objects, preferably described by few (< 10) attributes. Typically, obtaining high accuracies for high-dimensional data is difficult when relatively few objects (< 100 per class) are available. Many applications have such characteristics, especially those involving textual attributes. For example, email classification and text categorization tasks often involve thousands of attributes [14].

Case-based approaches for classification have been shown to be effective for some tasks with these characteristics, provided that suitable background knowledge is made available [15]. These approaches also outperform some other approaches when the features have not been carefully engineered [16]. Thus, we begin our investigations with a simple case-based classification approach, and will later compare it with others.

Case-based web services classification

Case-based classification proceeds as follows. To classify a new object, it reuses the classifications of previously classified objects (i.e., *cases*) that have characteristics similar to the new object. For example, each object in Table 1 is a case and the list of objects in the table constitute the *case base*. To assess the similarity of one case with another, the classifier uses a similarity metric. For example, the well known Euclidean distance metric can be used as a similarity function. The cases that are the most similar to the unclassified object are called its nearest neighbors. The classifier considers the classes of the k nearest neighbors from the case base when predicting the class label of an unclassified object. Training the classifier typically implies estimating the parameters of the similarity metric. Next, we describe the case-based approach we use for the Web Service classification task.

Web service classification in the AMB entails assigning one of two labels, "MetOc" or "non-MetOc", to a Web Service in question. The input to the classifier is a Web Service schema described using the WSDL [17] and the output is an associated label. The process of training the classifier on example cases is shown in Figure 3.

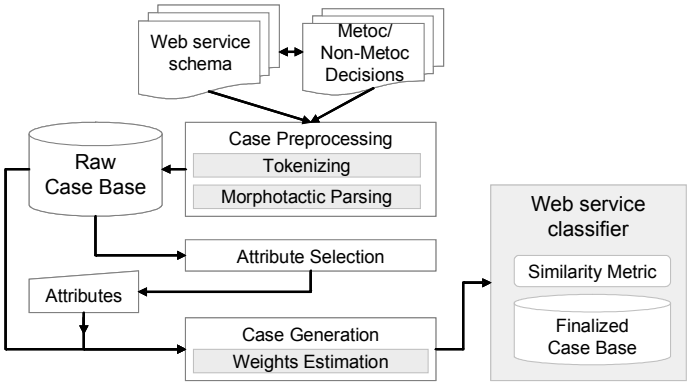

Fig. 3. Web Service classifier training process.

Case pre-processing: The WSDL describes the messages accepted by a Web Service and either contains or references an XML Schema. For classification, each WSDL must be converted into a case with attributes and values. We treat all the

element contents in the associated schema as a source of attributes. For example, an element in a schema may contain the enumerated value "waterTemperature". Its content can be directly used as an attribute. Alternatively, to reduce the sparseness of cases, it can be decomposed into its constituent terms. This is performed by a tokenization process, which decomposes such a string into its constituent words. For example, "waterTemperature" is decomposed into "water" and "temperature". Subsequently, a morphotactic parsing process further reduces words into their baseforms [18]. For example, the word "producer" is reduced to its baseform "produce". This approach allows us to reduce a Web Service schema to a bag of unique baseforms. Each baseform is a potential case attribute, where the frequency of its occurrence in a particular schema is its value. This is stored as a raw case in a preliminary case base. For each case, the decision of whether it is "MetOc" or "non-MetOc" is added as its class attribute.

Attribute selection: With potentially hundreds of example Web Services for classifier training, we expect to generate thousands of attributes. This poses a serious computational challenge to the classifier and can also adversely affect classification performance by introducing noisy and irrelevant attributes. For example, the attribute "http" may appear in all cases and provide no useful information to discriminate MetOc from non-MetOc Web Services. To counter this problem, we perform attribute selection, where a metric is used to select a subset of attributes with a potential to improve classification performance. Numerous attribute selection metrics exists, including mutual information, information gain, document frequency [19], and rough set methods [14]. We apply the information gain metric to select attributes in the Web Service Classifier.

Case Generation: After the attributes have been selected, each case must be indexed with the selected attributes and their corresponding weights must be computed. In this initial study, we use the information gain metric to calculate the weights applicable to the attributes. This results in a classifier that includes the finalized cases and the similarity metric.

After training is complete, the classification of a previously unknown Web Service proceeds as follows. A web service whose classification is unknown is submitted to the classifier. Case pre-processing and case generation processes are used to convert the Web Service schema into a case. This case is matched with the cases in the case base using the learned similarity metric and its k-nearest neighbors are retrieved. Their classes are then applied to the new case as follows. Each nearest neighbor votes on the decisions based on its classification. Each vote is weighted by the similarity of the voting neighbor. The classification label with the most (weighted) votes is assigned as the class of the new case. If the class assigned to the new case is the same as its actual class, then this is counted as a correct classification. Classifier performance is measured by the percentage of cases classified correctly.

Evaluation

We evaluated the Web Service Classifier. For our study, we implemented the classifier's preprocessor, attribute selector, and case generator. We obtained a set of 64 Web Services schemas from registries on the Web. Our meteorological subject matter expert then classified 26 of these schemas as MetOc relevant. We used a leave-one-out cross-validation (LOOCV) method to evaluate our classifier's performance, in which we repeatedly remove one case from the data set for testing and use the remaining cases to train the classifier. The classification accuracy for each test case is recorded using their respective trained classifier. This process of training and classification is repeated for each case in the set to determine the classifier's average classification accuracy.

The maximum classification accuracy of the Web Service Classifier was 93.75%, at k=5 and the number of attributes = 523 (out of maximum possible 1790). We used a genetic algorithm to search for the values of the parameters k and the number of attributes threshold used in the information gain feature selection algorithm. We used classification accuracy as the fitness function for the genetic algorithm .

CONCLUSION AND FUTURE DIRECTION

We described a novel method of automating the identification of MetOc Web Services within the context of an intelligent broker, the AMB. In this context, we described a case-based classification approach for Web Service identification. We reported the accuracy level achieved by our approach. In addition to autonomously identifying MetOc Web Services, the AMB will also be expected to independently match the user's data request to the correct method within the web service, to translate the user's request to the Web Service request, to dynamically invoke the method on the service, and to translate the Web Service response. These issues are more complex than Web Service identification. Whether classification approaches may prove beneficial in addressing these tasks is a focus of our future research. Additionally, as part of its mediation function, the AMB may also have to invoke multiple Web Services where the data required by the user is not readily available from a single service. Also significant to the end-user is the AMB's assessment of data confidence and reliability. We believe that current findings warrant additional work on the applicability of classification approaches to automating machine discovery and integration of Web Services.

ACKNOWLEDGMENTS

The authors would like to thank the Naval Research Laboratory's Base Program, Program Element No. 0602435N for sponsoring this research.

REFERENCES

[1] Director, Force Transformation, Office of the Secretary of Defense, "Network-Centric Warfare Creating a Decisive Warfighting Advantage", Washington, DC , Winter 2003. Cleared for Public Release by Department of Defense directorate for Freedom of Information and Security Review 04-S-0272.

[2] Paolucci, M., Soudry, J., Srinivasan, N., & Sycara, K., (2004), A Broker for OWL-S Web services, Proceedings of the AAAI Spring Symposium on Semantic Web Services.

[3]Andreas Heß and Nicholas Kushmerick, (2003) Automatically attaching semantic metadata to Web services, Proceedings of IJCAI.

[4] Andreas Heß and Nicholas Kushmerick, (2004) Machine Learning for Annotating Semantic Web Services, Proceedings of the AAAI Spring Symposium Semantic Web Services.

[5] XML Schema Tutorial, 2004, http://www.w3schools.com/schema/schema_intro.asp.

[6] Web Services Definition Language, 2004, http://www.perfectxml.com/WebSvc3.asp.

[7] Cerami, E., 2002, *Web Services Essentials*, O'Reilly and Associates, 2002.

[8] http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/

[9] http://www.w3.org/TR/2004/REC-owl-features-20040210/

[10] http://www.aifb.uni-karlsruhe.de/WBS/cte/ontologyengineering/

[11] http://protege.stanford.edu/overview/

[12] http://www.daml.org/services/owl-s/1.1/overview/

[13] Michie, D., Spiegelhater, D.J., & Taylor, C. (Eds.) (1994). Machine Learning, Neural and Statistical Classification, New York: Ellis Horwood.

[14] Gupta, K.M, Moore, P.G., Aha, D.W, & Pal, S.K. (2005). Rough-Set Feature Selection Methods for Case-Based Categorization of Text Documents, in S. K. Pal, S. Bandyopadhyay, & S. Biswas (Eds.) Lecture Notes in Computer Science (LNCS 3776), (pp. 792-798) : Heidelberg,Germany: Springer.

[15] Cain, T., Pazzani, M.J., & Silverstein, G. (1991). Using domain knowledge to influence similarity judgment. *Proceedings of the Case-Based Reasoning Workshop* (pp. 191--202). Washington, DC: Morgan Kaufmann.

[16] Aha, D.W. (1992). Generalizing from Case Studies: A Case Study, in D. Sleeman & P. Edwards (Eds.), Proceedings of the Ninth International Workshop on Machine Learning (ML92) (pp. 1-10) San Mateo, CA: Morgan Kaufmann.

[17] WSDL (2005), Web Service Description Language, Version 1.1, http://www.w3.org/TR/wsdl.

[18] Gupta, K.M., & Aha, D.W. (2004). RuMoP: A Morphotactic Parser, in R. Sangal & S.M. Bendre (Eds.), Proceedings of the International Conference on Natural Language Processing (ICON-2004), (pp. 280-284). Mumbai, India: Allied Publishers.

[19] Yang, Y., & Pederson, J. (1997). A comparative study of feature selection in text categorization. Proceedings of the Fourteenth International Conference on Machine Learning (pp. 412-420). Nashville, TN: Morgan Kaufmann.